PROJECT ABSTRACT

PDF Merger and Splitter Desktop App

A desktop-app concept for everyday PDF chores: merge multiple PDFs in a chosen order, split a document into validated page ranges, read real page counts from uploaded files, and download working output PDFs — all in one window. Delivered built-to-order with the working demo, source, report, PPT and viva kit.

01. Title

PDF Merger and Splitter Desktop App

02. Introduction / Background

Students constantly wrestle with PDFs: lab records scattered across files that must be submitted as one document, a 200-page ebook from which only one unit is needed, scanned assignments in the wrong order. Online PDF tools solve this but upload personal documents to unknown servers — a poor trade for academic work. A local desktop app keeps everything on the machine: drag in files, arrange them, merge; or pick ranges and split — with every operation validated before it runs.

03. Problem Statement

Merging and splitting PDFs is a weekly student chore handled by ad-ware sites or nothing at all. The project addresses this with a desktop utility: a merge view with file ordering and live page totals; a split view with page-range inputs validated against the document's real page count (including overlap detection); real page-count parsing for user-uploaded PDFs; and genuine downloadable output PDFs generated client-side — no server, no upload.

04. Objectives

- List PDF files with name, real page count and size; support adding more via upload.
- Parse page counts from uploaded PDFs by scanning the document structure.
- Allow reordering (up/down) and selective inclusion for merges.
- Validate split ranges against the true page count and reject overlaps/out-of-range values.
- Show animated progress for merge/split operations.
- Generate real, downloadable output PDFs with correct page counts.
- Keep an operation history log (newest first).
- Deliver the complete student kit: demo source, report, PPT and viva Q&A.;

05. Proposed Methodology

The app is a desktop-style window with sidebar navigation and three views. Uploaded files are read as text and page objects counted via the /Type /Page pattern — a reliable structural signal in real PDFs. The merge pipeline concatenates the selected files' page counts and renders progress; the split pipeline validates each range (1 ≤ start ≤ end ≤ total, no overlaps) before partitioning. Output files are produced by a hand-rolled minimal PDF writer that emits a valid PDF 1.4 with the exact page count, downloaded via Blob URLs. Every operation appends to the history log.

06. System Architecture / Working

Merge view: drop zone, file cards with checkboxes, reorder/remove controls, selection summary with total pages, merge button with progress bar, result card with download. Split view: document selector, total-page readout, dynamic range rows (add/remove), validation on split, per-part result cards each with its own download. History view: chronological operation log. The PDF writer builds catalog/pages/page objects with a correct xref table — the downloads open in any reader.

07. Hardware / Software Requirements

- Windows/Linux/macOS desktop (demo runs in any modern browser)
- Electron or Tauri (production desktop path); HTML/CSS/JS for the interactive demo
- No server, no network calls — all processing is local

08. Expected Outcomes

- A working merge/split utility with validated ranges and live totals.
- Real page-count parsing for user-uploaded PDFs.
- Genuine downloadable output PDFs with correct page counts.
- A session history log of all operations.
- A complete report, PPT and viva Q&A; on the PDF structure handling.

09. Applications

- Combining lab records, assignments and certificates into single submissions.
- Extracting units/chapters from large ebooks and notes.
- Offline PDF handling in labs with no internet.

10. Future Scope

- True content-preserving merge via a full PDF library (pdf-lib).
- Page rotation, deletion and reordering within a document.
- Compress and watermark operations.
- Batch folder processing with naming templates.